AD-A120 741 IRG - INTERACTIVE ROUTE GENERATOR: A NARRATIVE DESCRIPTION(U) GEORGIA INST OF TECH ATLANTA PRODUCTION AND DISTRIBUTION RESE.. F H CULLEN ET AL. FEB 82 1/1

UNCLASSIFIED PDRC-82-15 N00014-80-K-0709 F/G 9/2 NL

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A120741

Georgia Institute of Technology

PRODUCTION and DISTRIBUTION RESEARCH CENTER

"Original contains color plates: All DTIC reproductions will be in black and white"

SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING
GEORGIA INSTITUTE OF TECHNOLOGY
A UNIT OF THE UNIVERSITY SYSTEM OF GEORGIA
ATLANTA, GEORGIA 30332

82 10 08 026

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification

Distribution/
Availability Codes
Dist | Avail and/or Special
A

DTIC COPY INSPECTED 2

PDRC Report Series 82-1
February 1982

IRG - INTERACTIVE ROUTE GENERATOR: A NARRATIVE DESCRIPTION

by

Frank H. Cullen†
John J. Jarvis†
H. Donald Ratliff†

PDRC 82-15

Original contains color plates: All DTIC reproductions will be in black and white

†School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

This research was supported in part by the Office of Naval Research under Contract No. N00014-80-k-0709. Reproduction in whole or in part is permitted for any purpose of the U. S. Government.

I. INTRODUCTION

IRG is a prototype computer model which is designed to handle vehicle routing and delivery problems. This package employes an innovative interactive technique which combines the sophistication of mathematical models and advanced heuristic algorithms with state-of-the-art technology in color graphics and operator interfaces.

The system consists of a Chromatics CG Series color graphics terminal connected in real-time to a host main frame (CDC 760) computer. The human operator interacts with the color graphics terminal to direct the routing process at a supervisory level. The operator suggests route directions and strategic decisions to the computer, while the computer makes low-level tactical decisions and supplies information on higher level decisions to the human.

The operation of the system is facilitated through several user-friendly features. The graphic representation of the routing problem is clear and easy-to-use with intelligent use of colors and graphic figures to identify depots, demand points, and the routes which link them together. For example, the operator may choose to focus on one section of the geographical area. A "zoom" feature helps eliminate screen clutter and eases operator fatique. The graphic interface portion of IRG is entirely menu-driven with a simple tree-based menu calling sequence. An operator can thus be trained in a short period of time to use effectively all the powerful features of the system. Operator input is accomplished either through the keyboard or with a light pen if the graphics terminal is so equipped.

In the next section, we will discuss the basic characteristics of the problem IRG is designed to address. We will then present a general systems overview, followed by specific treatments of both the graphics interface and the underlying mathematical modelling code.

II. THE DELIVERY PROBLEM

The delivery (or vehicle routing) problem is one which occurs frequently in the real world. Examples of this problem include grocery chain supply and local small package delivery. Generally, there is one or more central depots with a fixed fleet of delivery vehicles which service populations of demand points. Each vehicle has a capacity (measured in weight and/or volume), while each demand point requires a certain amount of some commodity, (ies) (also measured in weight and/or volume).

The objective of a good solution to the delivery problem is to determine a set of routes - one for each vehicle - of least cost so that the weight and/or volume for each vehicle is not violated. Other restrictions, which are often imposed upon solutions to this problem, include a maximum route length, "time windows" on arrival and departure times for the demand points and requirements for loading/unloading times at the demand points and the central warehouse(s).

An example of a simple delivery problem having one depot, ten demand points and no time windows is given in Figure 1. Specific data for the problem in Figure 1 is given in Table 1. Since there is a total of 977 units demand, we must have at least 2 routes. A "good" solution with two vehicles is given by :†

1. D-8-7-3-1-4-D   Demand = 494, route length = 358
2. D-10-6-9-2-5-D   Demand = 483, route length = 281

This solution is illustrated in Figure 2.

---

†This is probably the optimal solution; however, this is difficult to verify even for such a simple example.

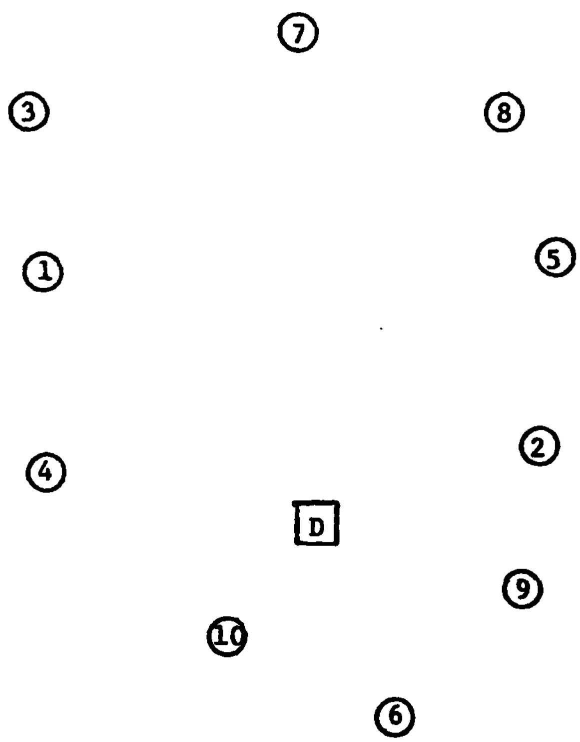

Figure 1. Delivery Example

Table 1

Delivery Problem Data

Avg Travel Speed = 2 grids/min

Vehicle Capacity = 500 units

Maximum Route Length = 360 minutes

Point Number	Location		Time(min)	(Units)
D	0	0		
1	-125	87	12	85
2	107	36	13	91
3	-146	120	5	32
4	-106	17	29	230
5	103	68	14	105
6	50	-64	9	59
7	-18	131	18	120
8	79	104	5	27
9	140	1	21	150
10	-20	-8	11	78

We have assumed here that the distance travelled by the vehicles is along a straight line between the demand points, and is translated into minutes. The travel time combined with the demand stop times yields the route times (lengths).

The solution illustrated in Figure 2 is very tight with respect to both the capacity and the route length constraints. If either the vehicle capacity were to be reduced 5% (to 475) or the maximum route length were to be reduced 5% (to 342 minutes) the solution of Figure 2 would not be feasible. In both situations, the addition of another vehicle might be necessary, and good solutions of the further restricted problems would be significantly different.

Simply adding one more vehicle to the sample problem makes good solutions more difficult to develop without a significant amount of trial and error. In general, for larger problems, some analytical support is required for the determination of good vehicle routes. The next section describes the general approach of IRG.

III. SYSTEM OVERVIEW

The IRG system can be described in two general environments; a physical environment, which is mainly concerned with what components are where; and the philosophical environment, which is principally concerned with methodological issues such as division of labor between the man and the machine and the underlying mathematical models. We will begin this section with a brief discussion of the philosophical environment with the intent of motivating the ensuing discussion of the system's physical components.

A. Methodological Issues

IRG is motivated by the notion of the vehicle delivery problem formulated as a set partitioning problem as in Balinski and Quandt [1]. Figure 3 gives a set partitioning formulation for the sample problem. Note that not all columns

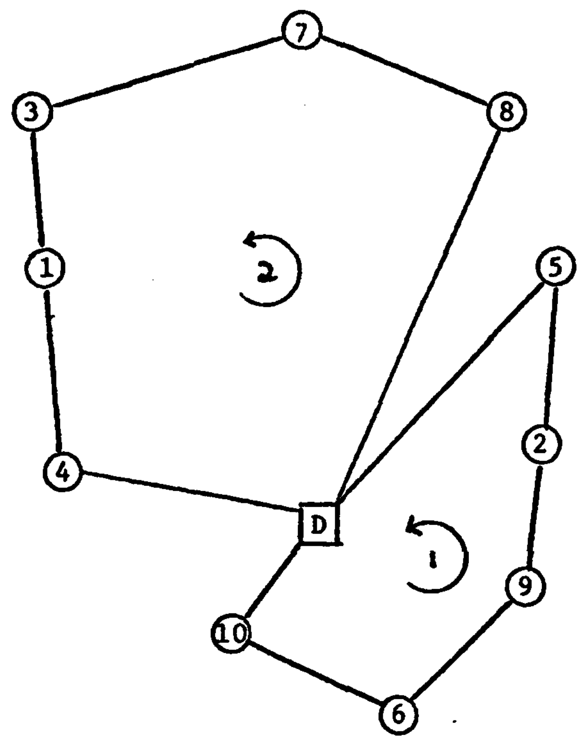

Figure 2. Solution to Example

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8		
Min Cost		304	263	233	255	281	319	347	358		
DEMAND POINT	1	1	1					1	1		1
	2				1	1	1				1
	3	1	1					1	1		1
	4	1	1	1				1	1		1
	5				1	1	1			=	1
	6			1	1	1		1			1
	7	1					1		1		1
	8						1		1		1
	9				1	1	1				1
	10		1	1		1		1			1

Figure 3. Partial Set Partition Formulation for the Delivery Example

(routes) have been generated. As discussed in Cullen, Jarvis and Ratliff [2], this procedure centers around the cooperation between the human operator and the machine with the purpose of generating feasible routes (representing feasible set partitioning problem columns) based upon "prices" associated with the individual demand points (and motivated by the dual variables of the linear programming relaxation). These columns provide the basis for new (hopefully improved) prices as the process repeats. Cullen, Jarvis and Ratliff [2] discuss the methodology of solving the delivery problem using set partitioning based heuristics. The current paper will not repeat this development; rather, it will concentrate on the overall design of the man-machine interactive system implementation. However, we will assume knowledge of certain terms (e.g. "savings," from the previous paper.)

The interactive impact of IRG on this column generation scheme is to place the human operator in control of the process by making supervisory (or strategic) decisions. These would be in the nature of determining when bad subordinate (or tactical) decisions are being made by the heuristic methods employed by the computer code and executing appropriate corrections. Figure 4 illustrates this general relationship between the human operator and the automatic portion of the IRG system.

It should be evident from Figure 4 that there is a high degree of interaction occurring between the human and the machine. Since the effectiveness of the IRG system is heavily dependent upon the effectiveness of the man-machine interaction, we will now examine some of the design considerations for the system which are concerned with achieving this interaction.

B. Screen Design

Consider the issue of :roblem ˜ .resentation. The human operator visualizes

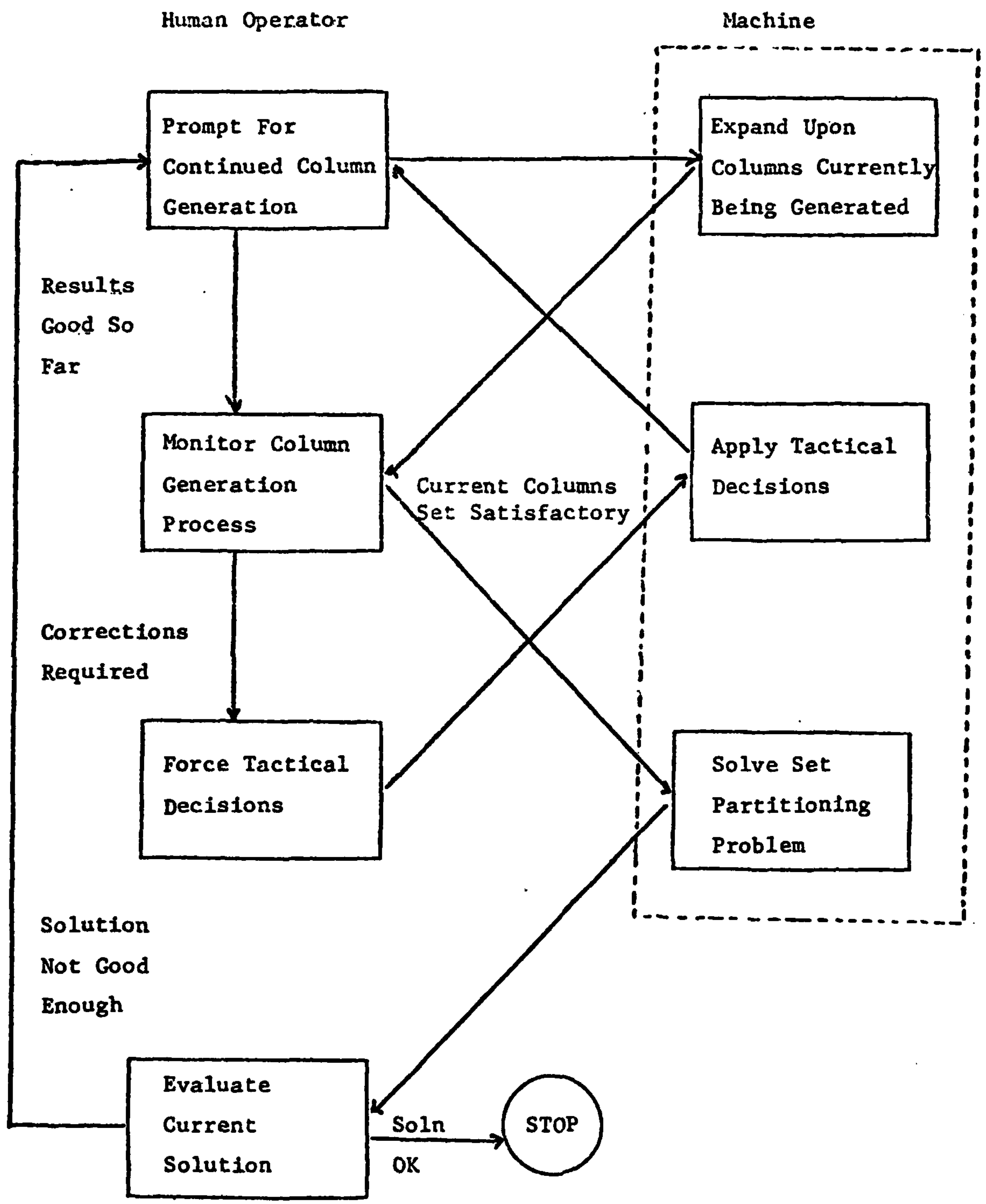

Figure 4. Human Operator-Machine Relationship

the vehicle delivery problem as a problem in which each of the components (warehouses, demand points, vehicles, routes) share a common environment with specific spatial relationships. So while a collection of demand points might be best stored as a list of coordinates and attributes in a computer, the human is much better at analyzing pictorial representation of the same data. IRG utilizes a color graphics computer for this latter purpose.

The actual display of the problem data on a color graphics screen employes judicious use of both colors and geometric figures. Simple graphics are selected for the warehouses and demand points. Green rectangles are used for warehouses and small circles for demand points - red when a demand point is not on a current route and yellow when it is. Notice that both color and shape are used to differentiate these two central problem elements. The rivers in the background map are dark blue - a most recessive color, while the highways which represent traffic arteries are white - a most dominant color. The general background for the problem representation is black, since black provides the best contrast for all other colors (taken as a group) and doesn't produce any glare or color convergence difficulties.

The actual routes are drawn in all the other available colors (except black and white). This allows the human operator yet another association to make with routes currently being constructed. Rather than forcing a numeric or code association with each route, the human operator can identify "the red route" or "the blue route" and avoid a confusing incoding/decoding process.

Since the human operator will also be required to control the process, we must also have the facility for the operator to efficiently interact with the system. IRG uses menus which allow prompting, by the machine, of available functions and produce a hierarchical software structure for the treatment

of the interaction on the machine side.

There are also numerous occasions when the information passing between the operator and the machine does not easily fit into either the graphical representation of the problem or the menu structure. Examples include initialization parameters, route summaries, and communications prompts. The IRG system allocates an area on the screen for this information. Figure 5 illustrates the general screen design for IRG. The menu area and the miscellaneous information area appear on the right side of the screen for two principal reasons:

1. The problem representation area is more square, which is desirable.
2. A right-handed person using a light pen can make a menu item selection without having to have his/her arm cross the field of view of the graphics area.

Light pen input is never accepted in the miscellaneous information area.

As a final note on screen layout, there are two other items which appear in appointed places on the screen. First, there is an indication of whether light pen input or keyboard cursor control is being used (these two options are mutually exclusive). This indication is made by the appearance either "KEYBOARD" or "LIGHT PEN" on a single line between the menu area and the miscellaneous information area. Second, since the human may not otherwise know when the machine is processing and when it is awaiting operator input, the prompt "MY TURN" or "YOUR TURN" appears (blinking) in the lower left-hand corner of the graphics area. Figure 6 shows an example of the screen layout.

C. Menu Structure

The menu structure of IRG is constructed so that those functions which share conceptual similarities or have functions which might be frequently used in conjunction with each other are together in the same menu group. Figure 7

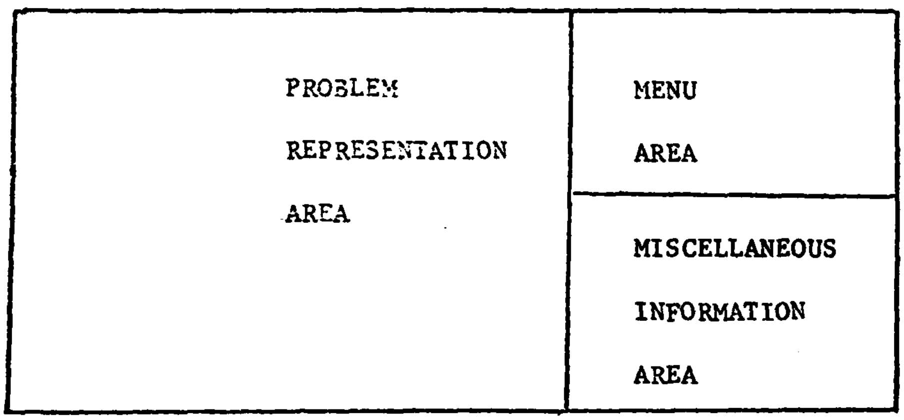

Figure 5. General IRG Screen Layout

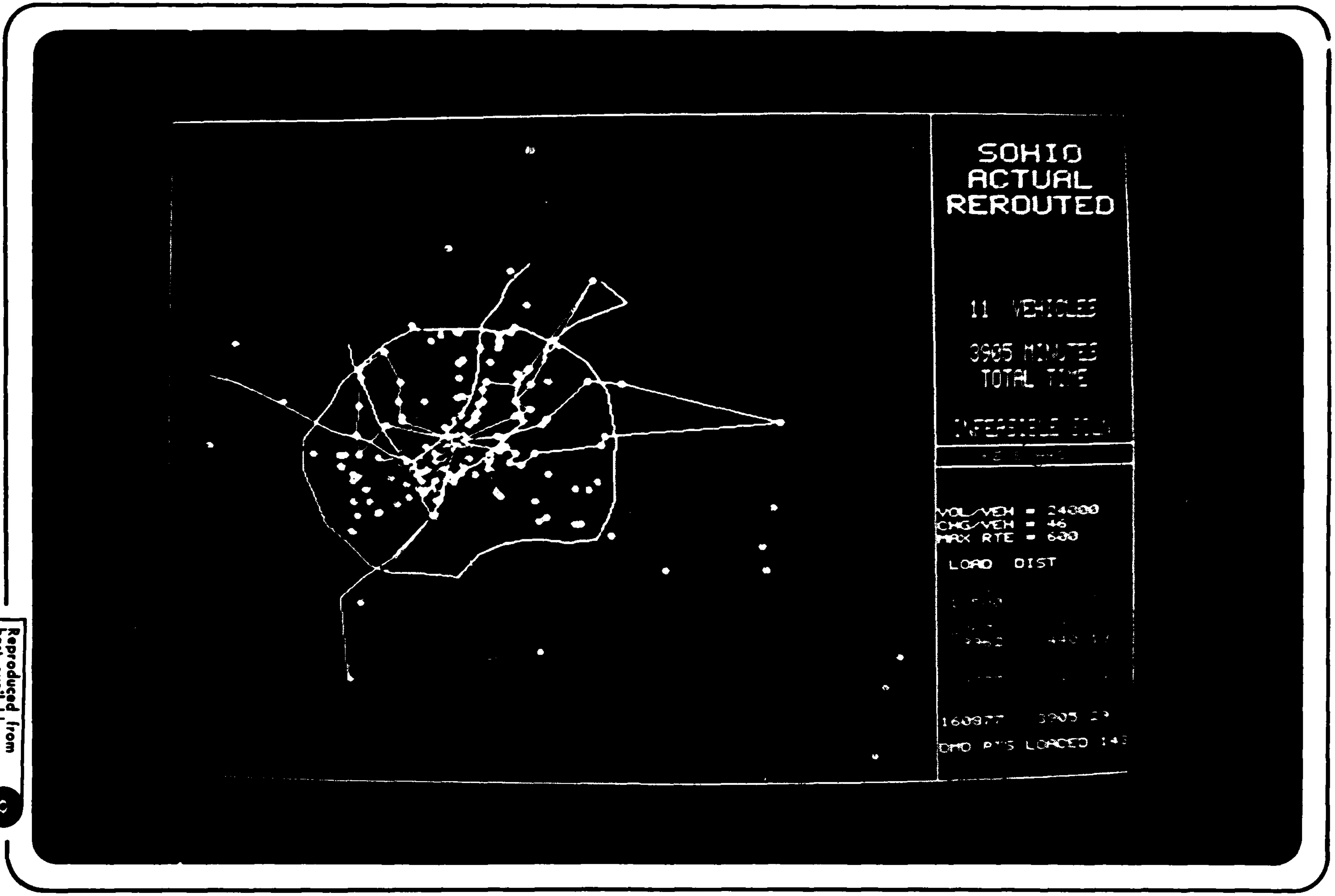

Figure 6. An Example of the IRG Graphics Screen Layout

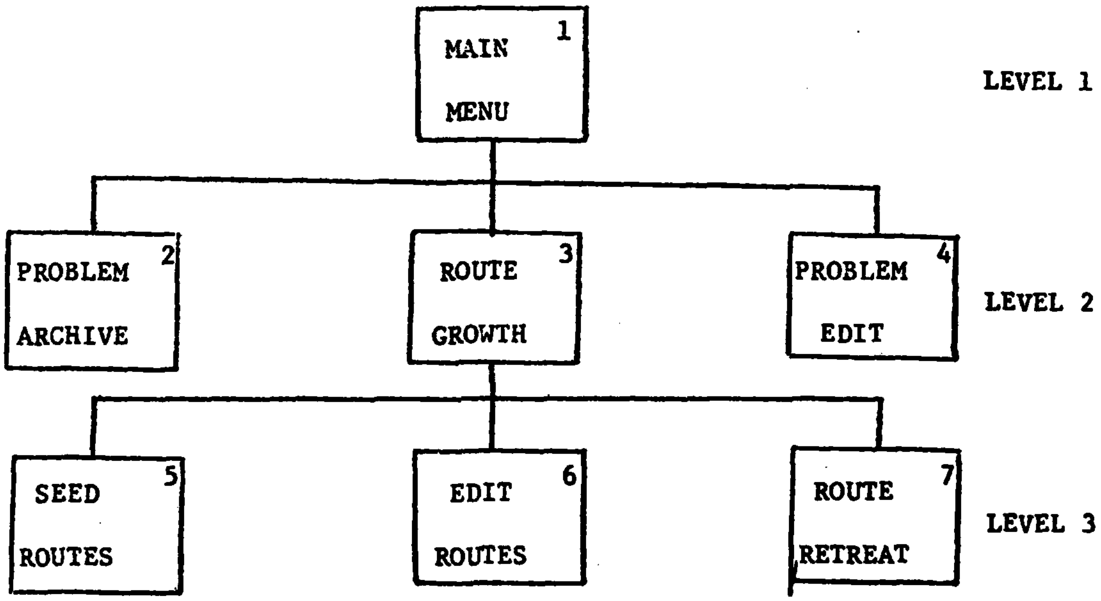

Figure 7: IRG Menu Group Configuration

illustrates the configuration of the currently available seven menu groups in a "call tree." The "call tree" arrangement indicates which menus can call other menus.

Each menu group includes functions of similar characteristics. This is so the operator does not have to constantly flip-flop between menu groups to access complementary functions. A menu group may be subordinate to some other "calling" menu group. Return to the "calling" menu group is enabled by an "Exit" menu selection within the subordinate menu group. For example, selecting "EXIT" while in the EDIT ROUTES menu group transfers control to the ROUTE GROWTH menu group. "EXIT" selection in the ROUTE GROWTH menu group transfer control to the MAIN MENU group.

Similarly, the MAIN MENU and ROUTE GROWTH menu groups each have three menu groups subordinate to them. Passage to these subordinate menu groups is made available by specific menu selections. For example, the MAIN MENU menu group includes items "ARCHIVE," "ROUTE GROWTH" and "PROBLEM EDIT." Selection of one of these options passes control to the corresponding subordinate menu.

We shall briefly discuss the functions of each of the seven menu subgroups. They are:

1. MAIN MENU - Level 1

   This menu group serves principally as an entry to the Level 2 menu groups. Provision is made here for further extensions to the IRG system - particularly the inclusion of set-partitioning and redistribution codes.

2. PROBLEM ARCHIVE - Level 2

   This menu group allows the operator to store and retrieve problems and solutions from the floppy disk.

3. ROUTE GROWTH - Level 2

This menu group serves as the central point for the route growth (column generation) procedures. Communications with the host machine (described later) are accomplished here. Access is also provided to the route - related subordinate menu groups SEED ROUTES, EDIT ROUTES, and ROUTE RETREAT.

4. PROBLEM EDIT - Level 2

This menu group contains the menu selections which are related to manipulations of the problem environment. This menu includes the software zoom and region editing capabilities.

5. SEED ROUTES - Level 3

The SEED ROUTES menu group provides the operator with the necessary options to initiate new routes in specified regions of the problem space.

6. EDIT ROUTES - Level 3

The ability to manually alter the tactical decisions made by the optimization model is provided in this menu group. The operator may delete or add items to routes, and merge or reroute current routes.

7. ROUTE RETREAT - Level 3

This menu group permits the operator to change strategic decisions. The underlying price multipliers may be altered, or the optimization process may be directed to retreat a number of past tactical decisions.

The aggregation of functions under the various menus is illustrated in Figure 8. The limitation of eight (8) choices under each menu group is motivated both by consideration of both clutter and readability in the small menu area in the upper right-hand corner of the graphics screen.

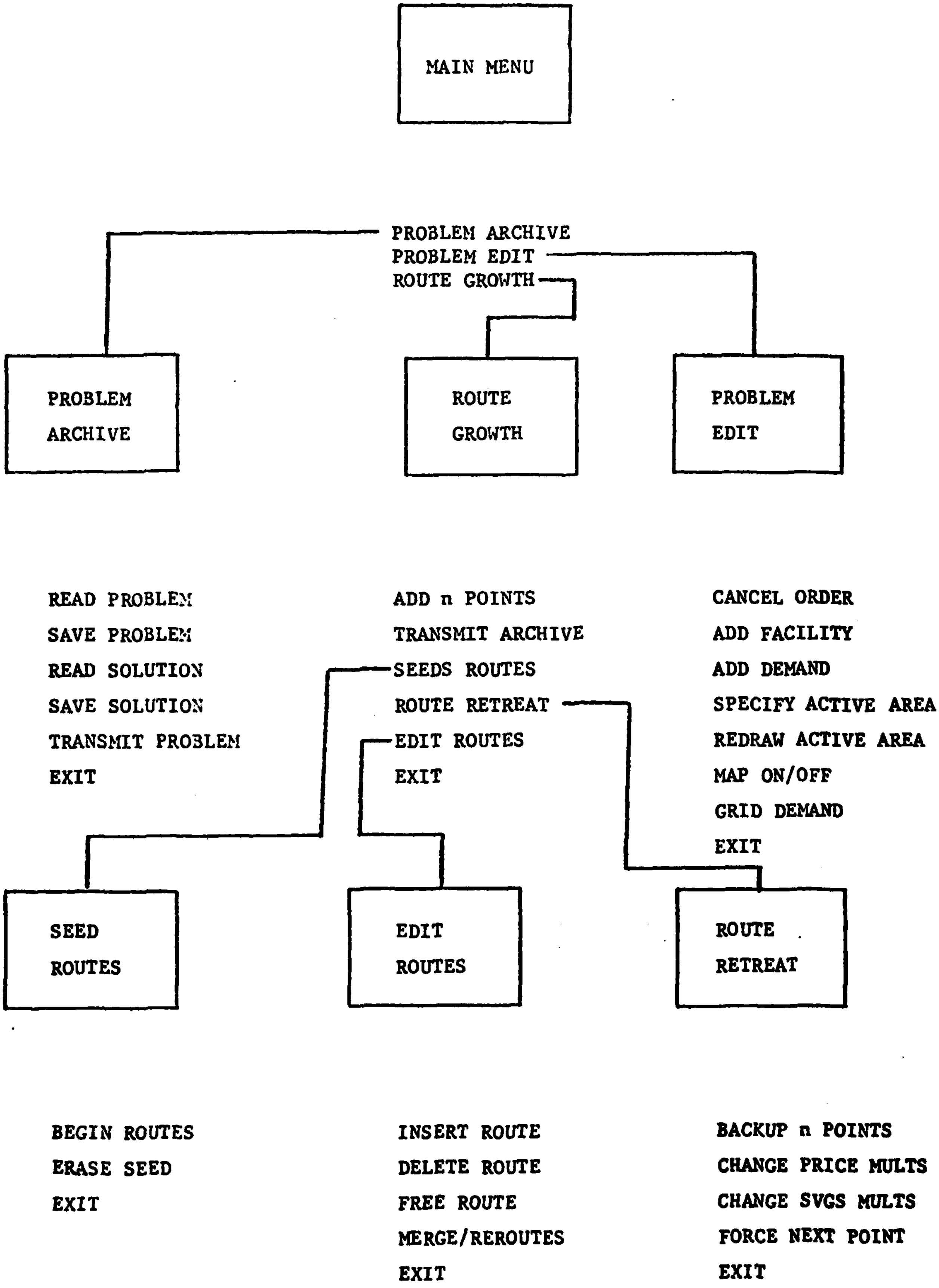

Figure 8. IRG Menu Schmatic

D. Division of Labor

We have already considered some of the mechanics of the man-machine interaction of IRG. We now consider the division of the labor between the color graphics unit and the host mainframe computer.

IRG was designed and implemented so that it's functions are separated into two parts: those functions which were best done locally on the color graphics terminal and those functions which were best performed by a computationally faster host mainframe computer. Figure 9 represents a schematic of the IRG system.

The functions of IRG which involve a substantial amount of computations are transmitted to the host mainframe computer. These functions are essentially savings computation and other tactical aids such as the travelling salesman code used by the MERGE/REROUTE option of the EDIT ROUTES menu group. The communication between the color graphics device and the host mainframe computer is handled logically as a simple transaction request. Specifically, there are a number of possible requests (transactions) which the color graphics device can make to the host. The host responds to this request with anything from a simple acknowledge (as in TRANSMIT ARCHIVE) to a series of transmissions (as in ADD n POINTS). Each transaction begins with a transaction code. A summary of these transaction codes is given in Table 2.

IV. IRG COLOR GRAPHICS SOFTWARE

IRG is implemented, at Georgia Tech, on a Chromatics CG 1999 color graphics computer. This unit has a high resolution (512 by 512) CRT screen, 8 colors, and is driven by an 8-bit Z80 microprocessor. The unit is a raster scan device with 4 color planes and many nice graphics primitive features (such as circles, complex fill, etc.). The unit also has a BASIC interpreter in ROM and 48k bytes

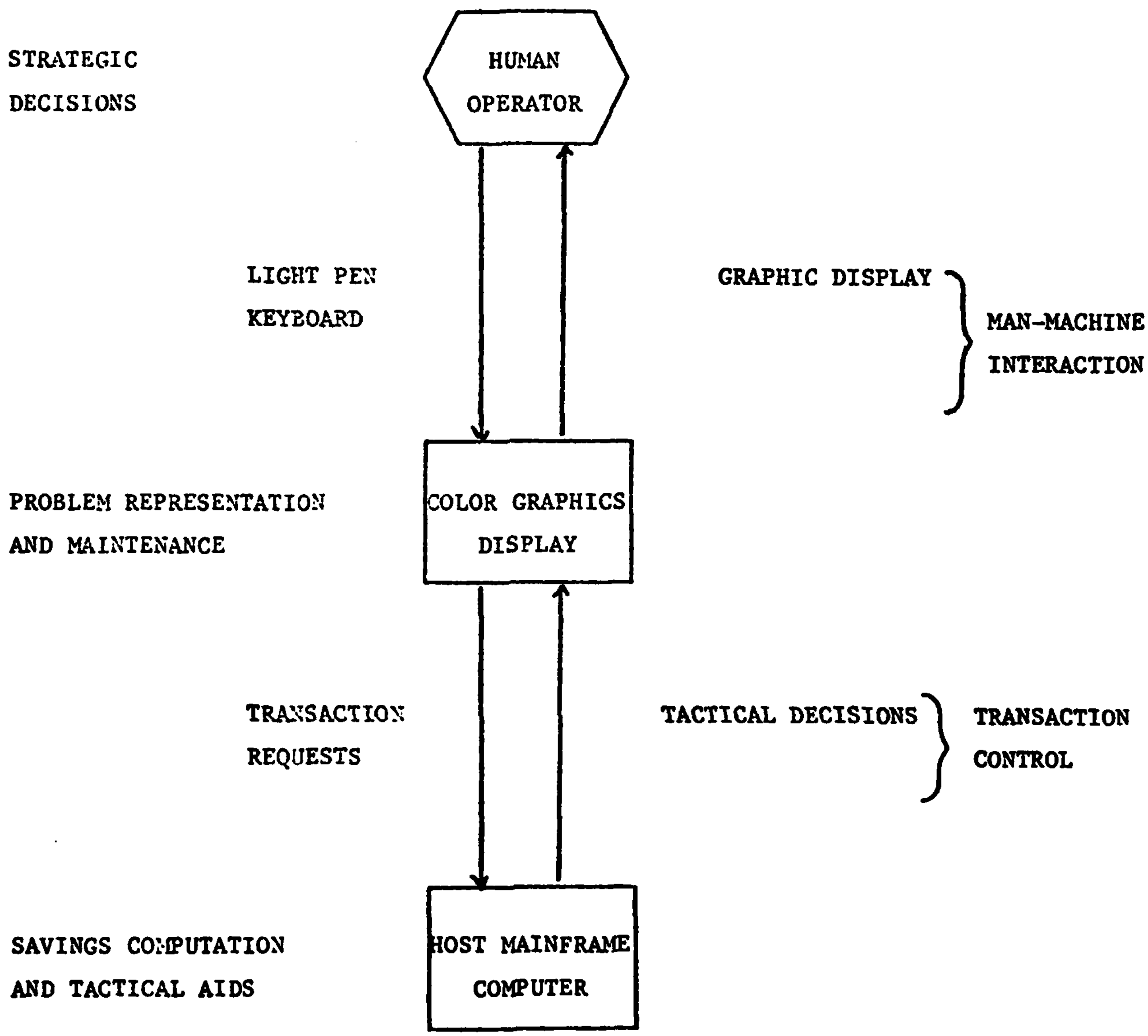

Figure 9. IRG System Schematic

Table 2

IRG Graphics Unit/Host Transaction Codes

TRANSACTION CODE	EXPLANATION	
11	Change Pricing Multiples	Growth Strategies
12	Algorithm Selection	
13	Change Savings Multiples	
30	Add n Points	Growth Tactics
31	Backup n Points	
40	Seed Routes/Transmit Archive	Synchronization
50	Read Problem From CG Unit	
60	Merge/Reroute (TSP)	Tactical Aids

of user-assessable RAM.

The Chromatics unit has a number of peripherals which IRG also uses. Among these are:

a. Two floppy disk drives - 8" single-sides, single-density

b. Light pen

c. Serial asynchronous communication port - for communication to the host mainframe computer

IRG is a development code and is written in BASIC rather than in Z80 assembler for reasons of simplicity, support, modification and portability. The BASIC interpreter requires about 20,000 bytes of RAM, leaving approximately 28,000 bytes for application programs and data. This storage limitation, together with the inherent slowness of the interpreter, make the Chromatics insufficient to handle the extensive computations required by IRG. Thus, the host computer is essential.

The color graphics software is designed to handle the man-machine interface. Manipulation of menus (as described in the last section) and the functions of graphical problem representation are the concern of the color graphics software. When communication with the host computer is required, the color graphics software also handles the request and transmission.

There are several data structures which are maintained locally on the color graphics unit. These include:

1. Coordinate points of demands and facilities: Distance measures are computed from this information.

2. Scaled coordinates for demands and facilities for fast access to, and identification of, demand points.

3. Demand amounts and delivery stop times

4. Route archives: A list structure representing the current vehicle routes.
5. Active window index set used to quickly identify and access demand and facility points within a zoomed region.

There is a library of routines which act upon these data structures. The overall program schematic for the color graphics software is illustrated in Figure 10. Notice that there is a unique menu protocol section for each menu group. Upon receiving an operator input, these protocol sections interpret the input and call upon elements of the macro library to carry out the required function. In brief, these library routines separate into six (6) groups:

1. Menu Service Routines: All actions which deal with menu service are centralized in this group. For example, the menus are numbered, and when control is passed from one menu group protocol to another, the macro to draw the new menu is called.
2. Interrupt Handling Routines: Program interrupts such as operator input or disk errors are trapped here. For example, when a menu group protocol is waiting for a light pen hit, one of these macros is called.
3. Window Handling Routines: The screen is divided into four parts, call "windows." To be sure that some part is not unintentionally erased, many protocols and several macros from other groups invoke window handling routines.
4. Communications Routines: Reception and trapping routines are located in this group which monitor the communication with the host computer.
5. Route Macros: The largest and most powerful group of the macro library is the route macros. Protocols from every part of the program call route macros to perform routines which are route oriented. The

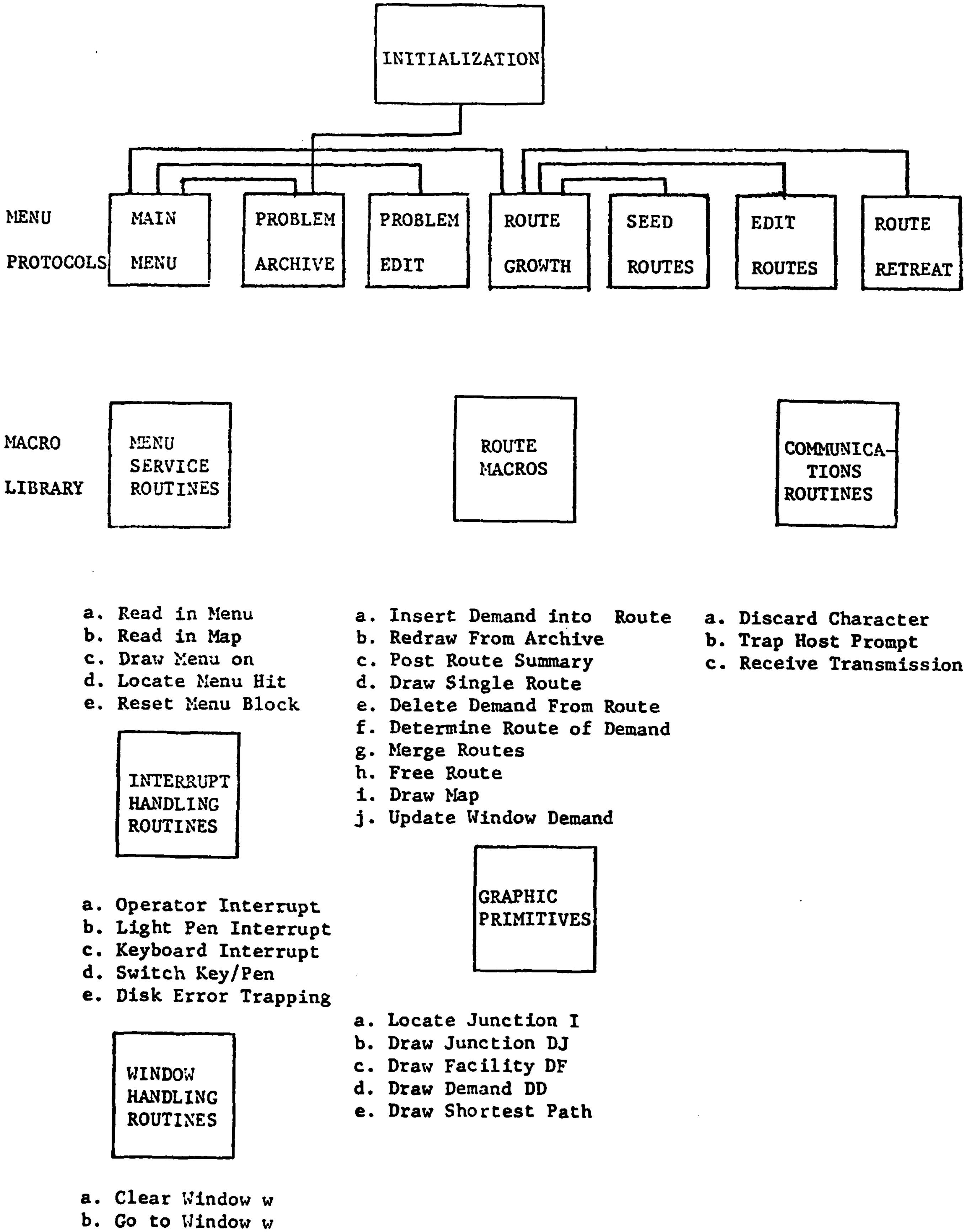

Figure 10. IRC - Graphic Terminal Software Schematic

route macros in turn make heavy use of the graphic primitives (described below) to isolate the representation of the problem from the data structures behind the computation.

6. Graphic Primitives (or Draw Macros): This is the lowest level of the representation software. This is the place where demands facilities and routes are drawn. The scaled coordinates are used to access and identify physical locations on the CRT screen.

V. IRG HOST COMPUTER SOFTWARE

The host computer, in which IRG has been implemented at Georgia Tech, is a CDC Cyber 760. The Source program is written in FORTRAN and communication occurs over phone lines to the Chromatics at rates of 110 to 2400 baud.

The host computer software is designed to handle the complex optimization-related computations required by IRG. The two principal data structures are the Savings Array and the Route Array. The bulk of the code is devoted to macros which deal with the maintenance of these two data structures. Figure 11 illustrates the configuration of the host computer code. As can be seen from the figure, the code separates into three (3) main groups:

1. Initialization: The problem coordinates and other pertinent information is read in from an external file. This eliminates the need for the problem data to be transmitted from the Chromatics - although that is also a feasible alternative.

2. Transaction Control: The role of the host computer is service to the color graphics unit. The host computer software awaits a transaction request. Upon receipt of a request, the host processes the request and returns either a simple acknowledgement or a series of transmissions supplying the requested information. During the processing of the

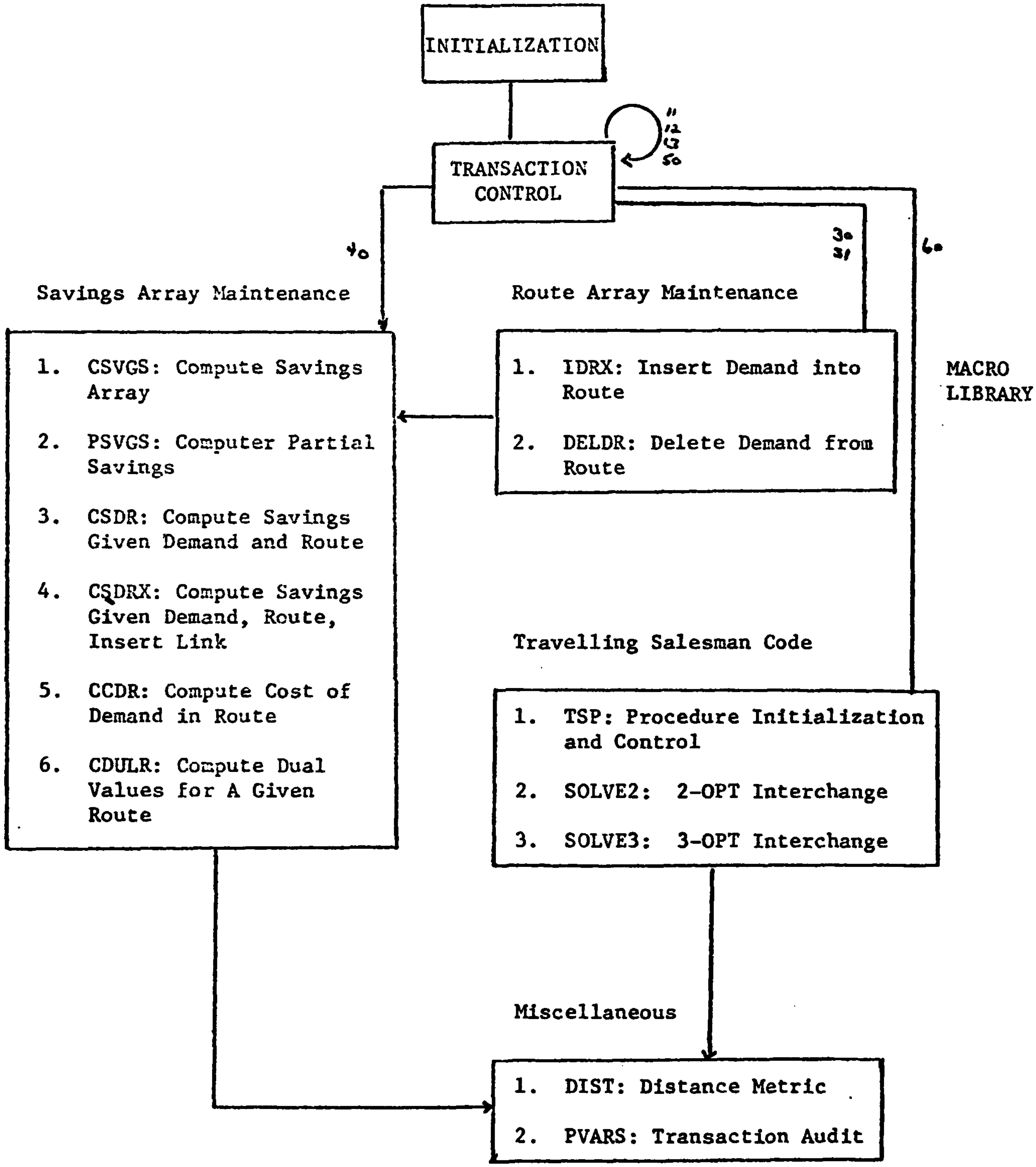

Figure 11. IRG - Host Computer Software Schematic

transaction, the transaction control may call one or more macros from the macro library. Figure 11 also contains indications of normal "call" paths into the macro library for certain transactions codes.

3. Macro Library: Similar to the color graphics software, the host computer code has a library of macro routines which operate on data structures. These macros may call each other (e.g., CSVGS calls CSDR which in turn calls CSDRX which calls DIST) or may be called by the transaction control code. The macro library divides into four (4) subgroups

   a. Savings Array Maintenance

   b. Route Array Maintenance

   c. Travelling Salesman (Tactical Aids)

   d. Miscellaneous (used throughout)

The various data structures are passed between program elements through the use of COMMON statements.

References

[1] Balinski, M., and R. Quandt, "On an Integer Program for a Delivery Problem," Operations Research, V. 12, pp. 303-304 (1964).

[2] Cullen, F., J. Jarvis and H. Ratliff, "Set Partitioning Based Heuristics for Interactive Routing," Networks, V. 12, No. 2, pp. 125-144 (1981).

END